

# Improved Hardness of Approximation for Stackelberg Shortest-Path Pricing

Patrick Briest \*

Sanjeev Khanna †

## Abstract

We consider the *Stackelberg shortest-path pricing problem*, which is defined as follows. Given a graph  $G$  with fixed-cost and pricable edges and two distinct vertices  $s$  and  $t$ , we may assign prices to the pricable edges. Based on the predefined fixed costs and our prices, a customer purchases a cheapest  $s$ - $t$ -path in  $G$  and we receive payment equal to the sum of prices of pricable edges belonging to the path. Our goal is to find prices maximizing the payment received from the customer. While Stackelberg shortest-path pricing was known to be APX-hard before, we provide the first explicit approximation threshold and prove hardness of approximation within  $2 - o(1)$ .

## 1 Introduction

The notion of *algorithmic pricing* encompasses a wide range of optimization problems aiming to assign revenue-maximizing prices to some fixed set of items given information about the valuation functions of potential customers [1, 11]. In a line of recent work the approximation complexity of this kind of problem has received considerable attention.

Without supply constraints, the very simple *single-price algorithm*, which reduces the search to the one-dimensional subspace of pricings assigning identical prices to all the items, achieves an approximation guarantee of  $\mathcal{O}(\log n + \log m)$ , where  $n$  and  $m$  denote the number of item types and customers, respectively [4, 6]. Corresponding hardness of approximation results of  $\Omega(\log^\varepsilon m)$  for some  $\varepsilon > 0$  are known to hold (under different complexity theoretic assumptions) even in the special cases that valuation functions are *single-minded* (items are *strict complements*) [10] or *unit-demand* (items are *strict substitutes*) [5, 7, 9]. In these cases, it is the potentially conflicting nature of different customers' valuations that constitutes the combinatorial difficulty of multi-dimensional pricing.

Another line of research has been considering so-called *Stackelberg pricing* problems [15], in which valuation functions are expressed implicitly in terms of some optimization problem. More formally, we are given a set of items, each of which has some fixed cost associated with it. In addition to these fixed costs, we may assign prices to a subset of the items. Given both fixed costs and prices, a single customer will purchase a min-cost subset of items subject to some feasibility constraints

---

\*Department of Computer Science, University of Paderborn, Germany. [patrick.briest@upb.de](mailto:patrick.briest@upb.de). Work done while the author was staying at Cornell University, supported by a scholarship of the German Academic Exchange Service (DAAD).

†Department of Computer and Information Science, University of Pennsylvania, USA. [sanjeev@cis.upenn.edu](mailto:sanjeev@cis.upenn.edu).

and we receive payment equal to the prices assigned to items purchased by the customer. As an example, we may think of items as being the edges of a graph and a customer aiming to buy a min-cost spanning tree, cheapest path, etc.

Clearly, as there is only a single customer in this type of problem, conflicting valuation functions can no longer pose a barrier for the design of efficient pricing algorithms. Yet, many Stackelberg pricing problems - and in particular the aforementioned spanning tree and shortest path versions - have so far resisted all attempts at improving over the single-price algorithm's logarithmic approximation guarantee. However, the best known hardness results to date only prove APX-hardness of both the spanning tree [8] and shortest path [12] cases without deriving explicit constants.

In this paper, we present the first explicit hardness of approximation result for the shortest path version of Stackelberg pricing, which we show to be hard to approximate within a factor of  $2 - o(1)$ . The result is based on a novel analysis of reduction that is quite similar to the ones previously described in [12] and [14].

## 1.1 Preliminaries

In the *Stackelberg shortest-path pricing problem* (STACKSP), we are given a directed graph  $G = (V, A)$ , a cost function  $c : A \rightarrow \mathbb{R}_0^+$ , a distinguished set of *pricable edges*  $\mathcal{P} \subset A$ ,  $|\mathcal{P}| = m$ , and two distinguished nodes  $s, t \in V$ . We may assign prices  $p : \mathcal{P} \rightarrow \mathbb{R}_0^+$  to the pricable edges. Given these prices, a consumer will purchase a shortest directed  $s$ - $t$ -path  $P^*$  in  $G$ , i.e.,

$$P^* = \operatorname{argmin} \left\{ \sum_{e \in P} (c(e) + p(e)) \mid P \text{ is } s\text{-}t\text{-path} \right\},$$

and we receive revenue  $\operatorname{rev}(p) = \sum_{e \in P^*} p(e)$ . We want to find a price assignment  $p$  maximizing  $\operatorname{rev}(p)$ .

Throughout the rest of this paper, we will w.l.o.g. only consider STACKSP instances for which  $c(e) = 0$  for all  $e \in \mathcal{P}$ , i.e., every edge is either pricable or fixed-cost, but never both.

## 2 Hardness of Approximation

We are going to show that STACKSP is quasi-NP-hard to approximate within a factor of  $2 - o(1)$ . The result is obtained by a refined analysis of a construction very similar to the one used previously in [12] and [14].

**Theorem 1** STACKSP cannot be approximated in polynomial time within a factor of  $2 - 2^{-\Omega(\log^{1-\varepsilon} m)}$  for any  $\varepsilon > 0$ , unless  $NP \subseteq DTIME(n^{\mathcal{O}(\log n)})$ .

### 2.1 Proof of Theorem 1

The proof of the Theorem is based on a reduction from the *label cover problem* (LABELCOVER), which is defined as follows. Given a bipartite graph  $G = (V, W, E)$ , a set  $L = \{1, \dots, k\}$  of *labels* and a set  $R_{(v,w)} \subseteq L \times L$  of satisfying label combinations for every edge  $(v, w) \in E$ , we want to find

a label assignment  $\ell : V \cup W \rightarrow L$  to the vertices of  $G$  satisfying the maximum possible number of edges, i.e., edges  $(v, w)$  with  $(\ell(v), \ell(w)) \in R_{(v,w)}$ . The following hardness result for LABELCOVER, which is an easy consequence of the PCP theorem [3] combined with Raz' parallel repetition theorem [13], is found, e.g., in the survey by Arora and Lund [2].

**Theorem 2** *For LABELCOVER on graphs with  $n$  vertices,  $m$  edges and label set of size  $k = \mathcal{O}(n)$  there exists no polynomial time algorithm to decide whether the maximum number of satisfiable edges is  $m$  or at most  $m/2^{\log^{1-\varepsilon} m}$  for any  $\varepsilon > 0$ , unless  $NP \subseteq DTIME(n^{\mathcal{O}(\log n)})$ .*

**Reduction:** Let an instance  $G = (V, W, E)$  with label set  $L = \{1, \dots, k\}$  as in Theorem 2 be given. Denote  $E = \{(v_1, w_1), \dots, (v_m, w_m)\}$ , where the ordering of the edges is chosen arbitrarily. Note that in our notation  $v_i, v_j$  for  $i \neq j$  may well refer to the same vertex (and the same is true for  $w_i, w_j$ ). For ease of notation we denote by  $R_i$  the satisfying label combinations for edge  $(v_i, w_i)$ .

We create a STACKSP instance as follows. For every edge  $(v_i, w_i)$  we construct a gadget as depicted in Fig. 1. Essentially, the gadget consist of a set of parallel pricable edges, one for each satisfying label assignment  $(\kappa, \lambda) \in R_i$  and an additional parallel fixed-cost edge of price 2.

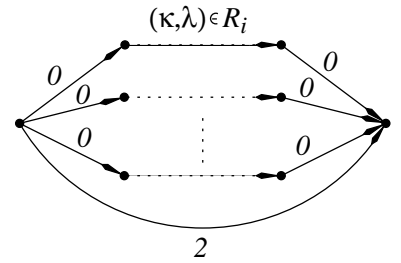


Figure 1: Gadget for an edge  $(v_i, w_i)$  in the label cover instance. Each pricable edge corresponds to one satisfying label assignment  $(\kappa, \lambda)$  to vertices  $v_i, w_i$ .

These gadgets are joined together sequentially (see Fig. 2). Let  $i < j$  and consider two pricable edges corresponding to label assignments  $(\kappa, \lambda) \in R_i$  and  $(\mu, \nu) \in R_j$ . We connect the endpoint of the first edge with the start point of the second edge with a *shortcut edge* of cost  $j - i - 1$ , if the two label assignments are conflicting, i.e., if either  $v_i = v_j$  and  $\kappa \neq \mu$  or  $w_i = w_j$  and  $\lambda \neq \nu$ . This construction is depicted in Fig. 2. Finally, we define the first node in the gadget corresponding to edge  $(v_1, w_1)$  and the last node in the gadget corresponding to  $(v_m, w_m)$  as nodes  $s$  and  $t$  the consumer seeks to connect via a directed shortest path. We will refer to the gadgets by their indices  $1, \dots, m$  and denote the pricable edge corresponding to label assignment  $(\kappa, \lambda)$  in gadget  $i$  as  $e_{i,\kappa,\lambda}$ .

**Completeness:** Let  $\ell$  be a label assignment satisfying all edges in  $G$ . We define a corresponding pricing  $p$  by setting for every pricable edge  $p(e_{i,\kappa,\lambda}) = 2$  if  $\ell(v_i) = \kappa$ ,  $\ell(w_i) = \lambda$  and  $p(e_{i,\kappa,\lambda}) = +\infty$  else.

The resulting shortest path from  $s$  to  $t$  cannot use any of the shortcut edges, because, as  $\ell$  is a feasible label assignment, out of any two pricable edges corresponding to conflicting assignments, one must be priced at  $+\infty$ . Consequently, no path using a shortcut edge can have finite cost. On the other hand, since  $\ell$  satisfies every edge, there is a pricable edge of cost 2 in each of the gadgets. It is then w.l.o.g. to assume that the consumer purchases the shortest path using the maximum possible number of pricable edges and, hence, total revenue is  $2m$ .

**Soundness:** Let  $p$  be a given pricing resulting in overall revenue  $m + c$  and let  $P$  denote the shortest path purchased by the consumer given these prices. We will argue that there exists a label assignment  $\ell$  satisfying  $c/4$  of the edges in  $G$ .

First note that w.l.o.g. any pricable edge that is not part of path  $P$  has price  $+\infty$  under price assignment  $p$ . In particular, this means that in every gadget  $i$  there is at most a single pricable edge with a finite price. We call this edge the *P-edge* of gadget  $i$ . We proceed by grouping gadgets into so-called *islands* as detailed below.

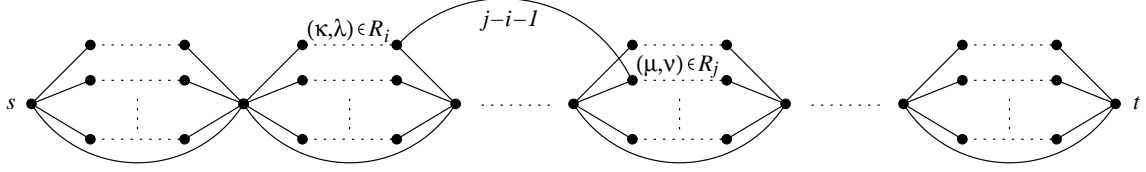


Figure 2: Assembling the edge gadgets into a STACKSP instance. Conflicting label assignments on two edges  $(v_i, w_i)$ ,  $(v_j, w_j)$  are connected by a shortcut of length  $j - i - 1$ . All edges are directed from left to right.

**Islands:** Let  $\sigma_1$  be the first gadget with a  $P$ -edge and call  $\sigma_1$  the *start point* of an *island*. Now for each  $\sigma_i$  find the maximum value of  $j > \sigma_i$ , such that gadget  $j$  has a  $P$ -edge and there exists a shortcut edge between the  $P$ -edges of gadgets  $\sigma_i$  and  $j$ . If such a  $j$  exists, define  $\sigma_{i+1} = j$ , else call  $\sigma_i$  an *end point*, let  $k > \sigma_i$  be the minimum value such that gadget  $k$  has a  $P$ -edge, define  $\sigma_{i+1} = k$  and call  $\sigma_{i+1}$  a start point. If no such  $k$  exists, call  $\sigma_i$  an end point and stop. Let  $\sigma_r$  be the end point of the final island. We call  $\sigma_1, \dots, \sigma_r$  the *significant gadgets*.

Note that by construction every gadget with a  $P$ -edge is covered by some *island*, i.e., the interval defined by some consecutive start and end points.

**Fact 1** Consider an island  $\sigma_\alpha, \dots, \sigma_\omega$ . Path  $P$  does not enter gadget  $\sigma_\alpha$  or exit gadget  $\sigma_\omega$  via a shortcut edge.

*Proof:* If  $P$  exits  $\sigma_\omega$  via a shortcut edge, then  $\sigma_\omega$  could not have been declared an end point. If  $\sigma_\alpha$  is entered via a shortcut edge, this shortcut must originate from a gadget  $i < \sigma_\alpha$  which lies within the preceding island. As  $P$  cannot bypass the endpoint of the preceding island via a shortcut,  $i$  must in fact be the end point  $\sigma_{\alpha-1}$  and so  $\sigma_\alpha$  could not have become a start point.  $\square$

Consider now a single island  $\sigma_\alpha, \dots, \sigma_\omega$ . By  $\ell_i$  we denote the length of the shortcut edge between gadgets  $\sigma_i$  and  $\sigma_{i+1}$  for  $\alpha \leq i \leq \omega - 1$ . Furthermore, by  $in_i$  and  $out_i$  we refer to the lengths of the shortcut edges used by path  $P$  to enter and exit gadget  $\sigma_i$ , respectively, and set them to 0 if no shortcuts are used. From Fact 1 above it follows that  $in_\alpha = out_\omega = in_{\alpha+1} = 0$ . See Fig. 3 for an illustration.

For  $\alpha \leq i \leq \omega$  let the cost of path  $P$  between shortcut edges  $out_i$  and  $in_{i+1}$  be  $r_i + c_i$ , where  $r_i$  denotes the cost due to pricable edges and  $c_i$  the cost due to fixed-cost edges, respectively. We are going to bound the expression  $p_{\sigma_i} + r_i$ . We note that  $\ell_\omega = 0$ , since by the fact that gadget  $\sigma_\omega$  is an endpoint, no shortcut edge connects its  $P$ -edge to the  $P$ -edge of another gadget. Similarly, we have  $r_\omega = 0$ , since path  $P$  does not use pricable edges between islands, as we have argued before.

Path  $P$  crosses the end node of the  $P$ -edge in gadget  $\sigma_i$  (node  $v_2$  in Fig. 3) and the start node of the  $P$ -edge of gadget  $\sigma_{i+1}$  (node  $v_4$  in Fig. 3) for  $\alpha \leq i \leq \omega - 1$ . The total cost of path  $P$  between these two vertices is  $out_i + r_i + c_i + in_{i+1}$ . An alternative path  $P_1$  is obtained by replacing this part of  $P$  with the shortcut edge of length  $\ell_i$  between  $\sigma_i$  and  $\sigma_{i+1}$ . By the fact that  $P$  is the shortest path we have  $out_i + r_i + c_i + in_{i+1} \leq \ell_i$  and, thus,

$$r_i \leq \ell_i - out_i - in_{i+1} \quad \text{for } \alpha \leq i \leq \omega, \quad (1)$$

where the bound on  $r_\omega$  follows from the fact that for  $i = \omega$  all summands in the above expression are 0. Similarly, the cost of path  $P$  between the start node of the shortcut edge into gadget  $\sigma_i$  (node

$v_1$  in Fig. 3) and the end node of the shortcut edge exiting  $\sigma_i$  (node  $v_3$  in Fig. 3) is  $in_i + p_{\sigma_i} + out_i$  for  $\alpha \leq i \leq \omega$ . We obtain an alternative path  $P_2$  by taking only fixed cost edges of cost 2 to bypass both shortcuts and gadget  $\sigma_i$  at total cost  $2(in_i + out_i + 1)$ . Again, since  $P$  is the shortest path, we get  $in_i + p_{\sigma_i} + out_i \leq 2(in_i + out_i + 1)$ , or

$$p_{\sigma_i} \leq 2 + in_i + out_i \quad \text{for } \alpha \leq i \leq \omega. \quad (2)$$

Combining (1) and (2) yields

$$p_{\sigma_i} + r_i \leq 2 + \ell_i + in_i - in_{i+1} \quad \text{for } \alpha \leq i \leq \omega. \quad (3)$$

Finally, we have

$$\sum_{i=\alpha}^{\omega} (p_{\sigma_i} + r_i) \leq \sum_{i=\alpha}^{\omega} (2 + \ell_i + in_i - in_{i+1}) \quad (4)$$

$$= 2(\omega - \alpha) + \sum_{i=\alpha}^{\omega} \ell_i + in_{\alpha} - in_{\omega+1} = 2(\omega - \alpha) + \sum_{i=\alpha}^{\omega} \ell_i. \quad (5)$$

Recall that  $\sigma_1, \dots, \sigma_r$  denote the significant gadgets across all islands. Assume now that there is a total number  $I$  of islands with start and end points  $\sigma_{\alpha(1)}, \sigma_{\omega(1)}, \dots, \sigma_{\alpha(I)}, \sigma_{\omega(I)}$ . Summing over all islands we get that overall revenue of price assignment  $p$  is bounded by

$$\sum_{j=1}^I \sum_{i=\alpha(j)}^{\omega(j)} p_{\sigma_i} + r_i \leq \sum_{j=1}^I \left( 2(\omega(j) - \alpha(j)) + \sum_{i=\alpha(j)}^{\omega(j)} \ell_i \right) \leq 2(r - 1) + m,$$

where the last inequality follows from the fact that  $\alpha(j) = \omega(j - 1) + 1$  for  $2 \leq j \leq I$ ,  $\omega(I) = r$  and  $\sum_{i=1}^r \ell_i \leq m$ , since all shortcuts defining the  $\ell_i$  are disjoint. Thus, we have  $m + c \leq 2(r - 1) + m$ , or  $r \geq c/2 + 1$ .

Now consider the  $P$ -edges of the  $\lceil r/2 \rceil$  gadgets  $\sigma_1, \sigma_3, \sigma_5, \dots$  and their corresponding label assignments  $(\kappa_i, \lambda_i)$ . By definition, there are no shortcut edges between the  $P$ -edges of any of these gadgets and, thus,  $(\kappa_1, \lambda_1), (\kappa_3, \lambda_3), \dots$  define a non-conflicting label assignment satisfying at least  $\lceil r/2 \rceil \geq c/4$  edges in  $G$ . (Labels not defined by  $(\kappa_1, \lambda_1), (\kappa_3, \lambda_3), \dots$  can be chosen arbitrarily.)

Finally, consider a label cover instance as in Theorem 2 and the path pricing instance resulting from our reduction above. If all edges can be satisfied, maximum path pricing revenue is  $2m$ . If no label assignment satisfies more than  $m/2^{\log^{1-\varepsilon} m}$  edges, maximum path pricing revenue is bounded by  $(1 + 4/2^{\log^{1-\varepsilon} m})m$ . This finishes the proof of Theorem 1.

### 3 Tightness

We briefly mention that our analysis is tight in the following sense. It is easy to check that by assigning price 1 to all pricable edges we can make sure that w.l.o.g. the shortest  $s$ - $t$ -path uses a pricable edge in each of the gadgets and, thus, we obtain revenue  $m$ . Since maximum possible revenue is bounded above by  $2m$  (there is an  $s$ - $t$ -path of that cost that does not use any pricable edges), it follows that it is trivial to achieve approximation guarantee 2 on the instances resulting from our reduction.

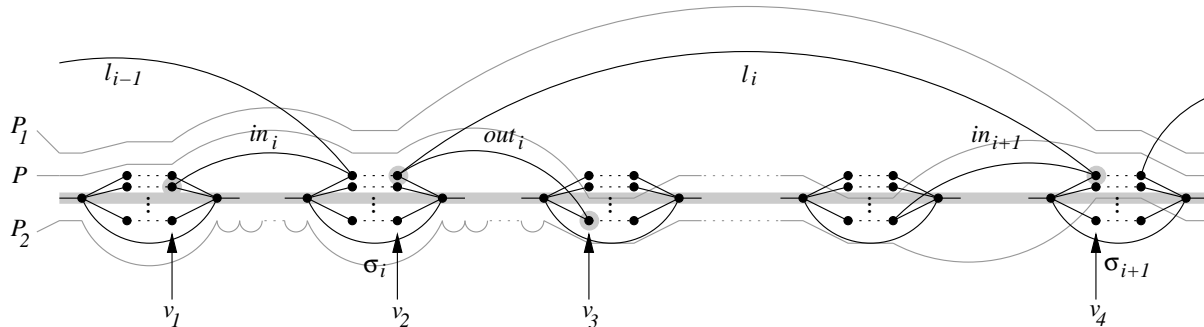


Figure 3: Two consecutive significant gadgets  $\sigma_i, \sigma_{i+1}$  inside one island. The length of the shortcut edges used to enter and exit gadget  $\sigma_i$  (defined as 0 if no such shortcut exists) are denoted as  $in_i$  and  $out_i$ , respectively.

## 4 Conclusions

We have proven the first explicit approximation threshold for any Stackelberg pricing problem. Still, the approximation threshold for this kind of problem in general - and the shortest path version in particular - is far from settled. The following questions seem to constitute fertile ground for future research:

- Can we prove super-constant hardness of approximation results for any kind of Stackelberg pricing problem?
- Is it possible to achieve a better than logarithmic approximation guarantee for the Stackelberg shortest path pricing problem? Is there an interesting restricted set of graphs on which constant approximation factors are possible?

## References

- [1] G. Aggarwal, T. Feder, R. Motwani, and A. Zhu. Algorithms for Multi-Product Pricing. In *Proc. of 31st ICALP*, 2004.
- [2] S. Arora and C. Lund. Hardness Of Approximations. In *Approximation Algorithms for NP-hard Problems*, PWS Publishing Company, 1996.
- [3] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof Verification and Hardness of Approximation Problems. *Journal of the ACM*, **45**(3):501–555, 1998.
- [4] N. Balcan, A. Blum, and Y. Mansour. Item Pricing for Revenue Maximization. In *Proc. of 9th EC*, 2008.
- [5] P. Briest. Uniform Budgets and the Envy-Free Pricing Problem. In *Proc. of 35th ICALP*, 2008.
- [6] P. Briest, M. Hoefer, and P. Krysta. Stackelberg Network Pricing Games. In *Proc. of 25th STACS*, 2008.
- [7] P. Briest and P. Krysta. Buying Cheap is Expensive: Hardness of Non-Parametric Multi-Product Pricing. In *Proc. of 18th SODA*, 2007.

- [8] J. Cardinal, E. Demaine, S. Fiorini, G. Joret, S. Langerman, I. Newman, and O. Weimann. The Stackelberg Minimum Spanning Tree Game. In *Proc. of 10th WADS*, 2007.
- [9] J. Chuzhoy, S. Kannan and S. Khanna. Network Pricing for Multicommodity Flows. *Unpublished manuscript*, 2007.
- [10] E.D. Demaine, U. Feige, M.T. Hajiaghayi, and M.R. Salavatipour. Combination Can Be Hard: Approximability of the Unique Coverage Problem. In *Proc. of 17th SODA*, 2006.
- [11] V. Guruswami, J.D. Hartline, A.R. Karlin, D. Kempe, C. Kenyon, and F. McSherry. On Profit-Maximizing Envy-Free Pricing. In *Proc. of 16th SODA*, 2005.
- [12] G. Joret. Stackelberg Network Pricing is Hard to Approximate. *CoRR*, *abs/0812.0320*, 2008. <http://arxiv.org/abs/0812.0320>.
- [13] R. Raz. A Parallel Repetition Theorem. *SIAM Journal on Computing*, **27**, 1998.
- [14] S. Roch, G. Savard, and P. Marcotte. An Approximation Algorithm for Stackelberg Network Pricing. *Networks*, 46(1): 57–67, 2005.
- [15] H. von Stackelberg. *Marktform und Gleichgewicht (Market and Equilibrium)*. Verlag von Julius Springer, Vienna, 1934.